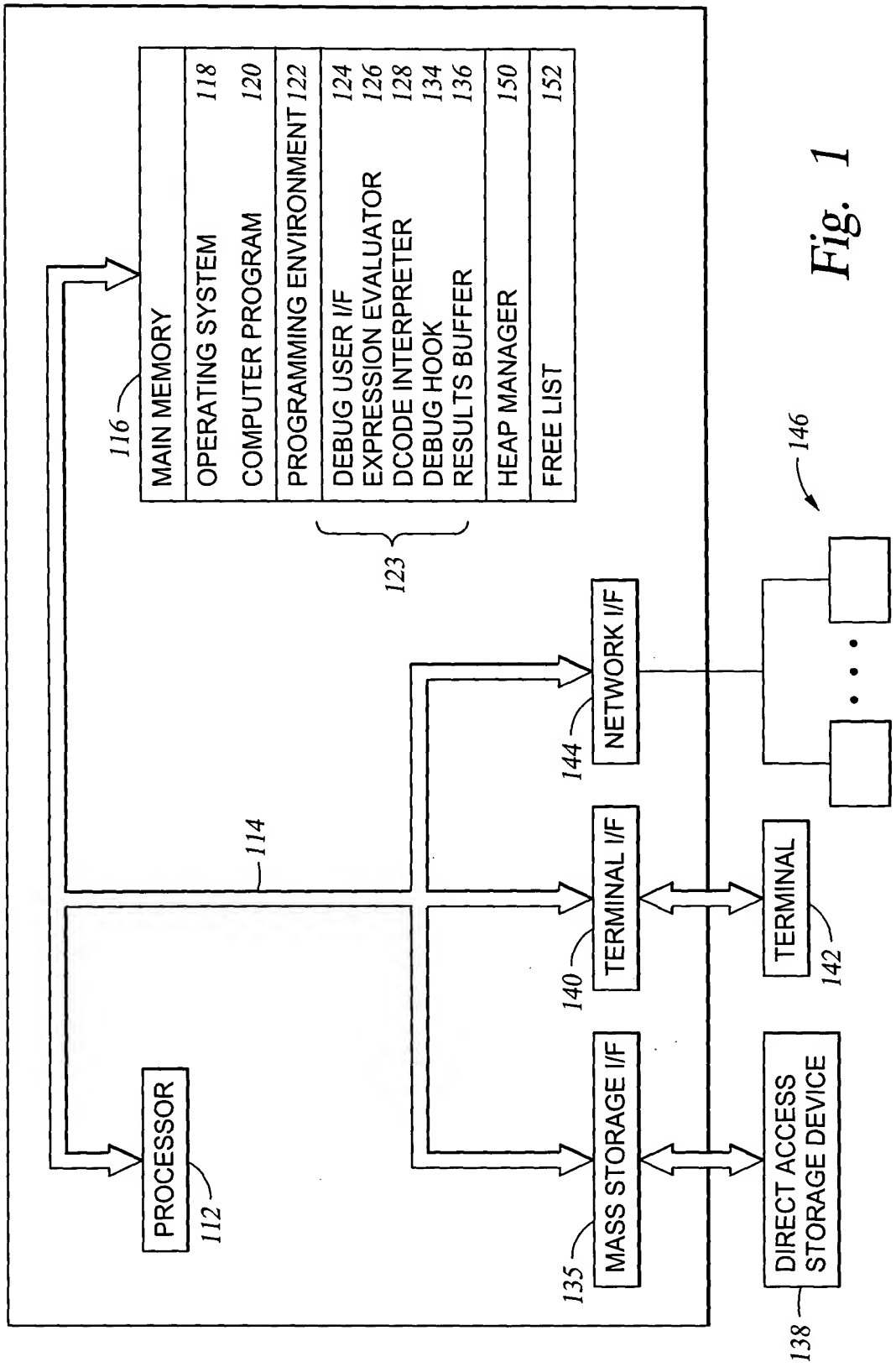
110
MAIN MEMORY
116
OPERATING SYSTEM 118
COMPUTER PROGRAM 120
PROGRAMMING ENVIRONMENT 122
DEBUG USER I/F 124
EXPRESSION EVALUATOR 126
DCODE INTERPRETER 128
DEBUG HOOK 134
RESULTS BUFFER 136
HEAP MANAGER 150
FREE LIST 152
123
146
114
PROCESSOR
112
NETWORK I/F
144
TERMINAL I/F
140
TERMINAL
142
MASS STORAGE I/F
135
DIRECT ACCESS STORAGE DEVICE
138

Fig. 1

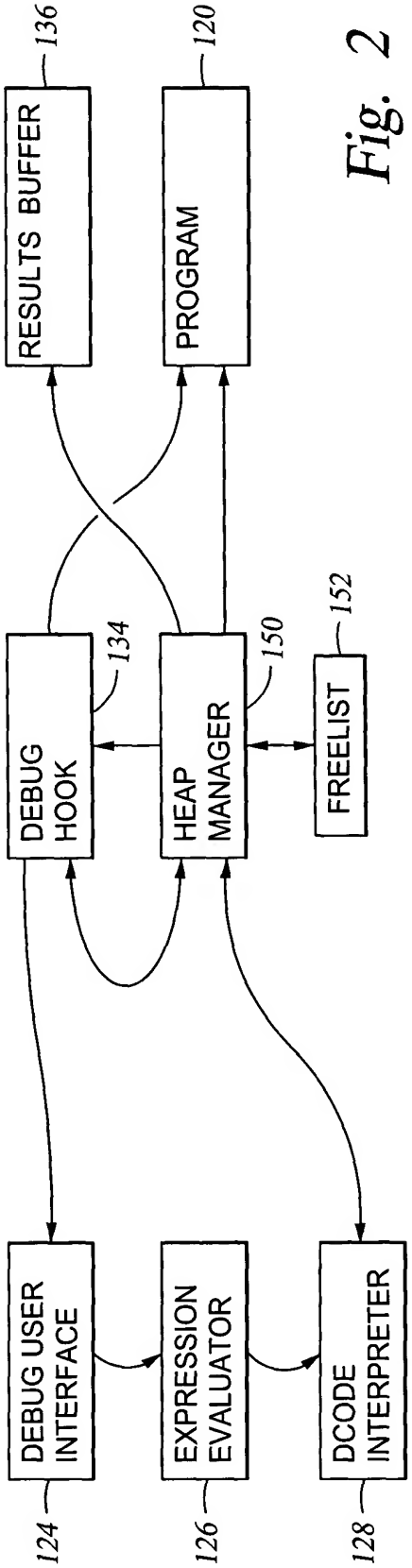

Fig. 2

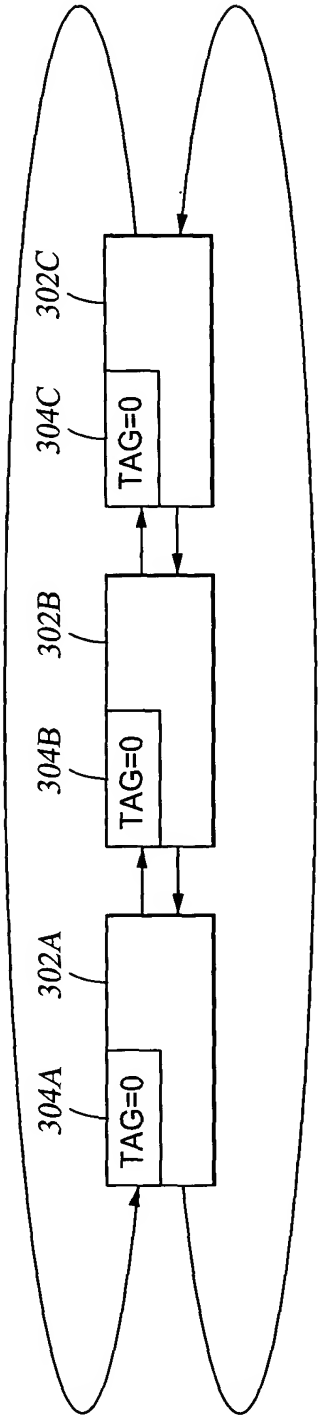

Fig. 3

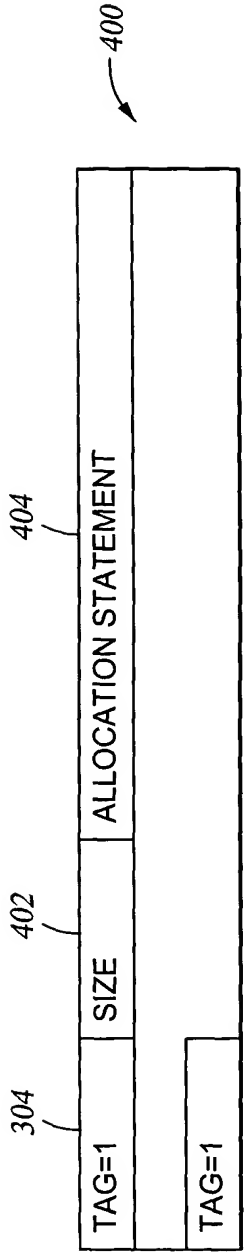

Fig. 4

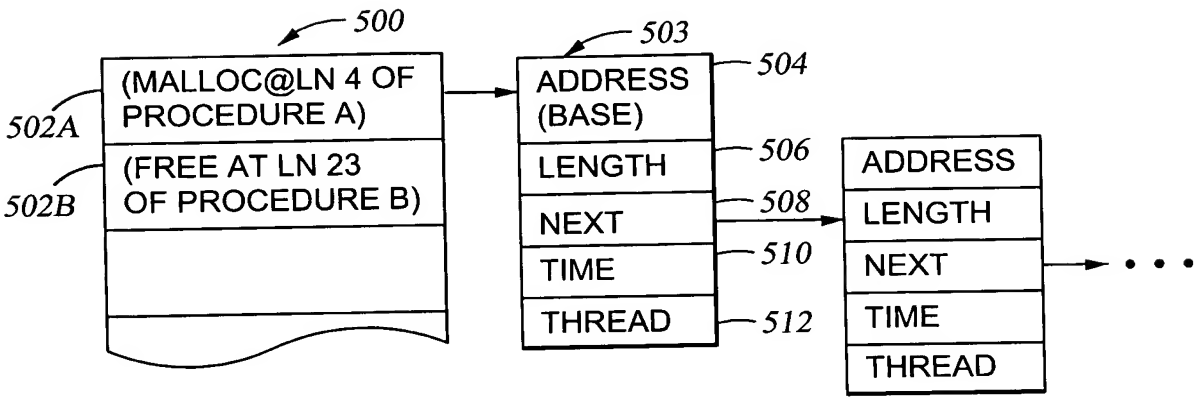

Fig. 5

600

```
001  # INCLUDE <STDLIB.H>
002 CHAR * COPY ( CHAR * P )
003 {
004    CHAR  *PTR;
005
006    PTR = MALLOC ( STRLEN (P) + 1 ) ;
007
008    STRCPY  ( PTR, P) ;
009
>>010    RETURN PTR;
011 }
```

Fig. 6

700

==>ALLOCHIST PTR

706 708 710

>ALLOCHIST PTR

PROC/LINE	LENGTH	PROC/LINE
COPY/006	4	F002/026
COPY/006	30	F002/026
PRINTF/987	128	PRINTF/1005
FOO1/5	48	

704

Fig. 7

802 ALLOCATOR STATEMENT	804 LIMIT	806 COUNT
	0	
	0	
	40K	40K
	10K	11K

800

Fig. 8

900

```
==> UPSTGIMT  6   4000

   001  # INCLUDE <STDLIB.H>
   002 CHAR * COPY (CHAR *P)
   003 {
   004     CHAR * PTR;
   005
>>006 PTR = MALLOC (STRLEN  (P) + 1) ;
   007
   008     STRCPY ( PTR,P ) ;
   009
   010     RETURN PTR;
   011 }

==>
UPPER STORAGE LIMIT OF 4000 EXCEEDED AT LINE 006
```

904

Fig. 9

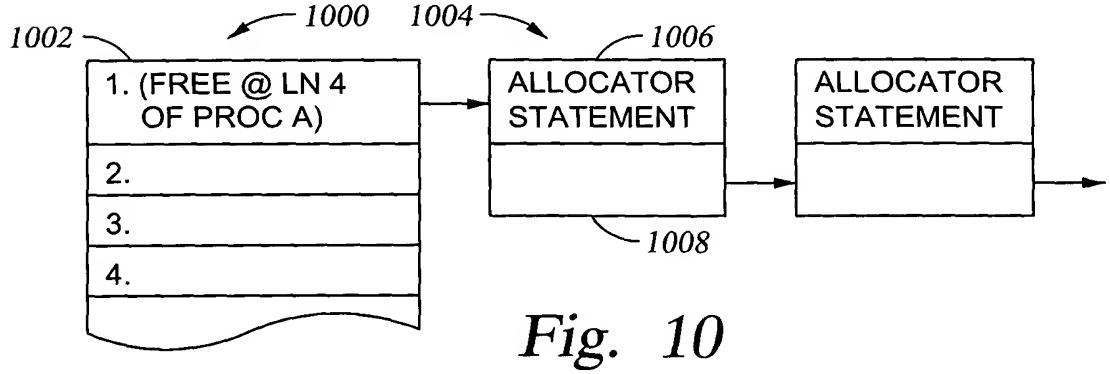

Fig. 10

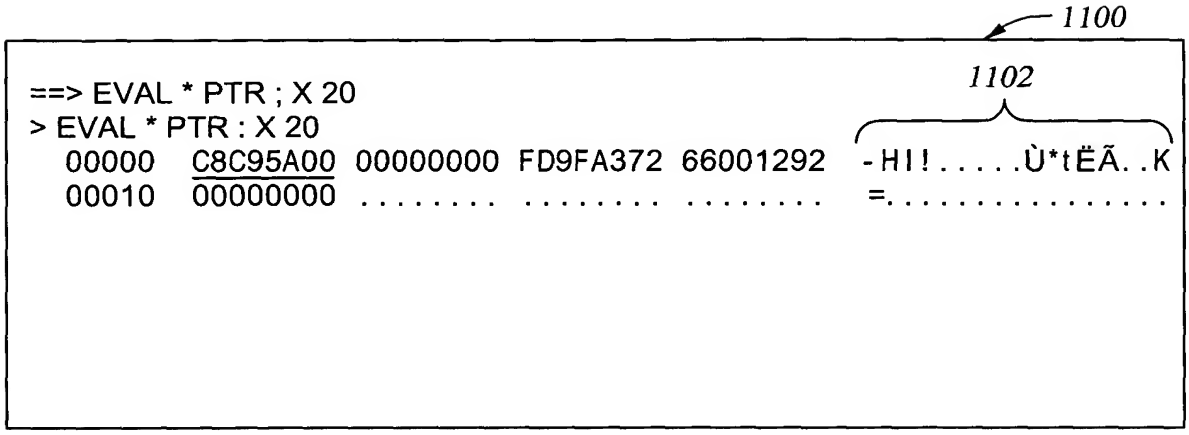

Fig. 11

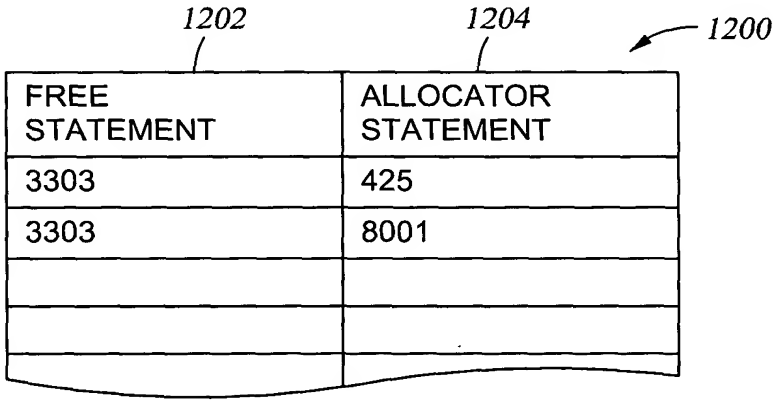

Fig. 12

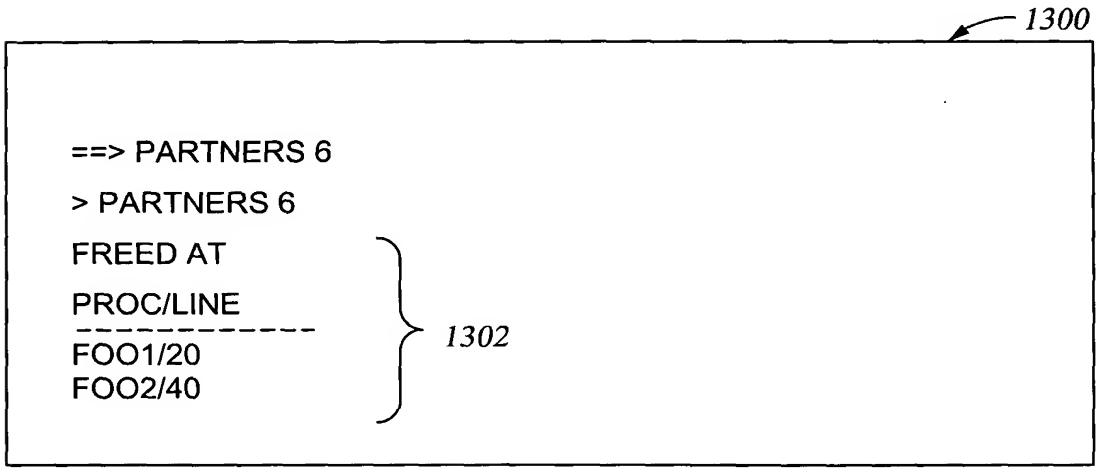

Fig. 13

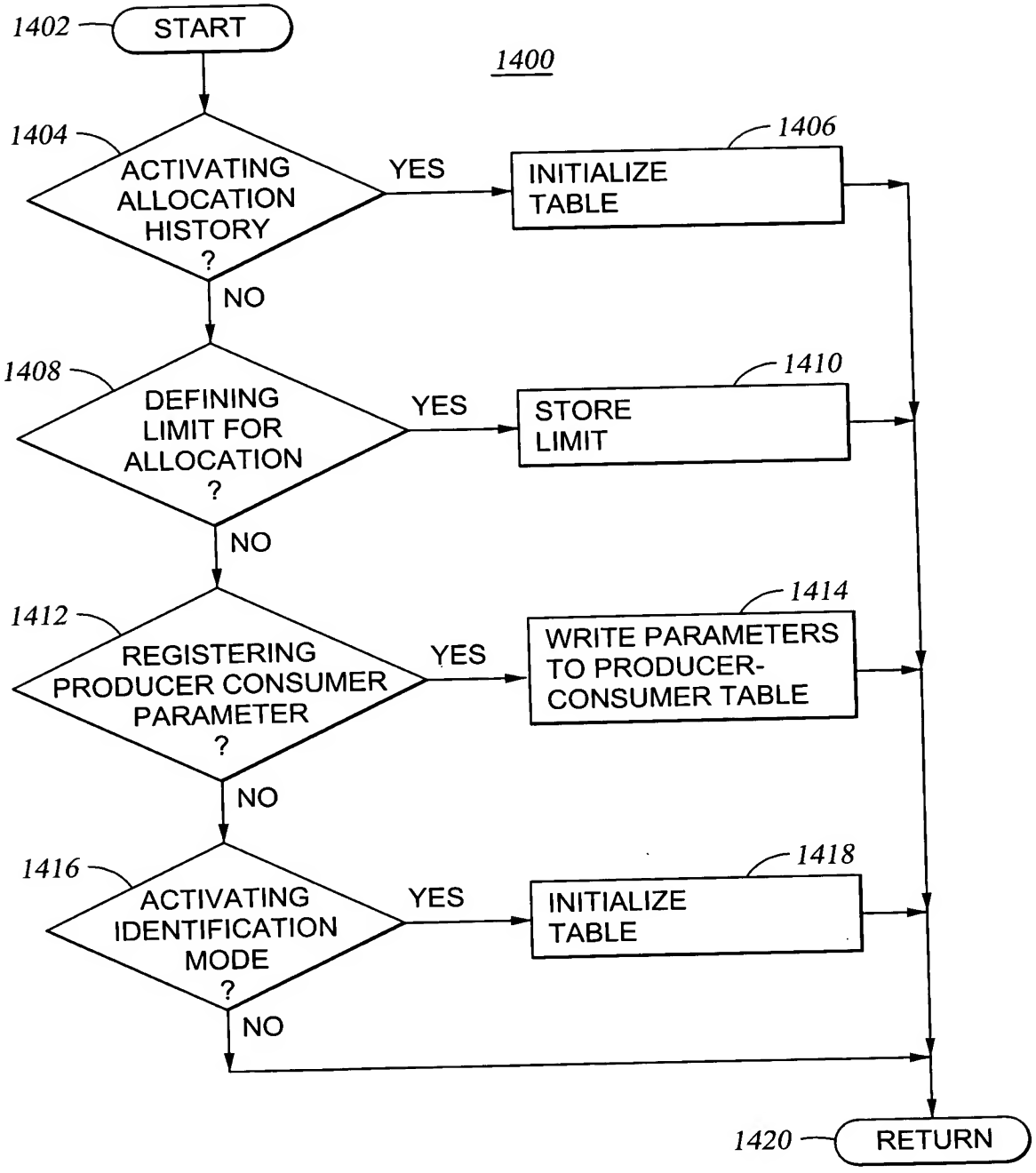
1402
START
1400
1404
ACTIVATING ALLOCATION HISTORY ?
YES
1406
INITIALIZE TABLE
NO
1408
DEFINING LIMIT FOR ALLOCATION ?
YES
1410
STORE LIMIT
NO
1412
REGISTERING PRODUCER CONSUMER PARAMETER ?
YES
1414
WRITE PARAMETERS TO PRODUCER-CONSUMER TABLE
NO
1416
ACTIVATING IDENTIFICATION MODE ?
YES
1418
INITIALIZE TABLE
NO
1420
RETURN

Fig. 14

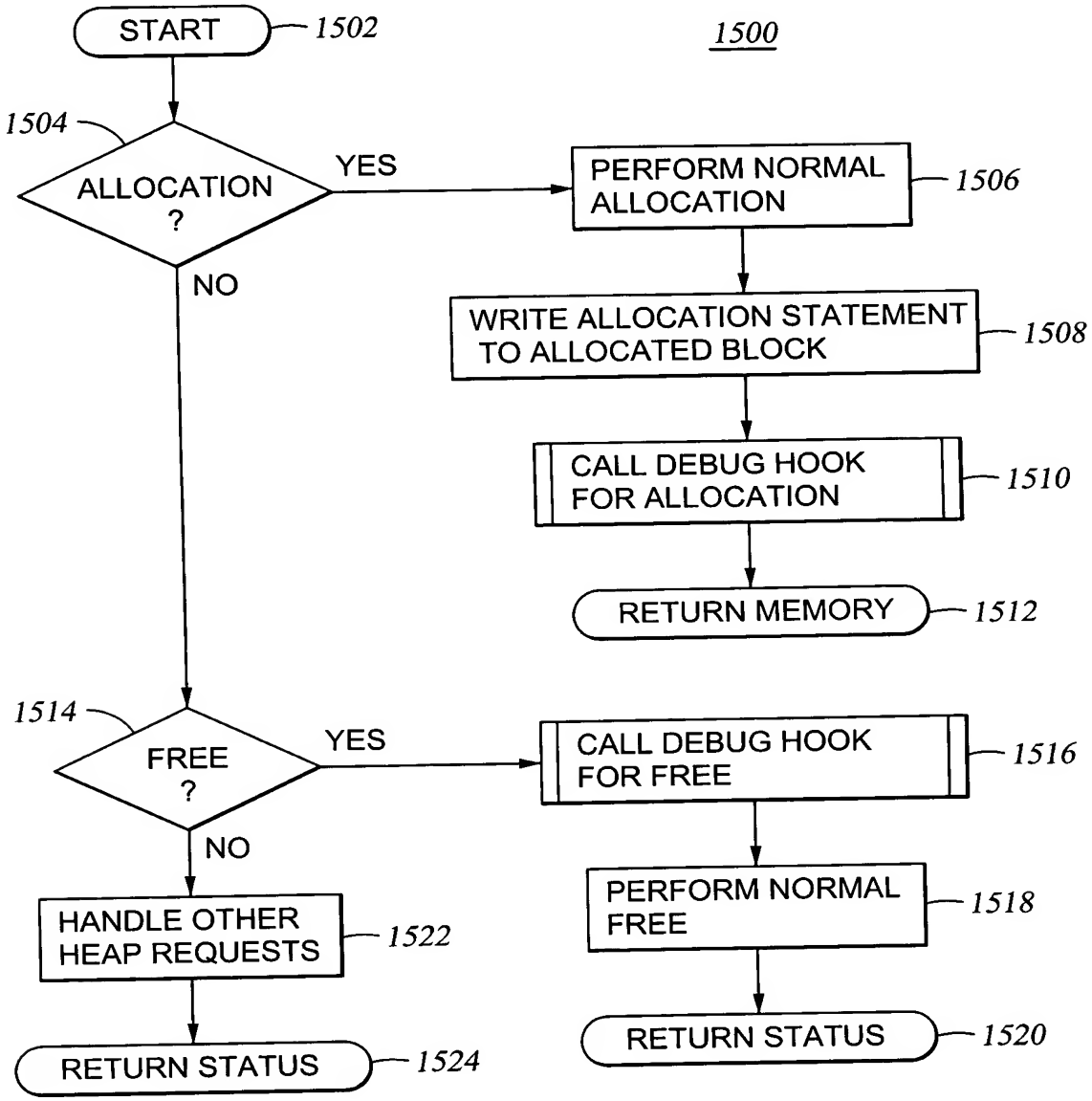
START
1502
1500
1504
ALLOCATION ?
YES
PERFORM NORMAL ALLOCATION
1506
NO
WRITE ALLOCATION STATEMENT TO ALLOCATED BLOCK
1508
CALL DEBUG HOOK FOR ALLOCATION
1510
RETURN MEMORY
1512
1514
FREE ?
YES
CALL DEBUG HOOK FOR FREE
1516
NO
PERFORM NORMAL FREE
1518
HANDLE OTHER HEAP REQUESTS
1522
RETURN STATUS
1520
RETURN STATUS
1524

Fig. 15

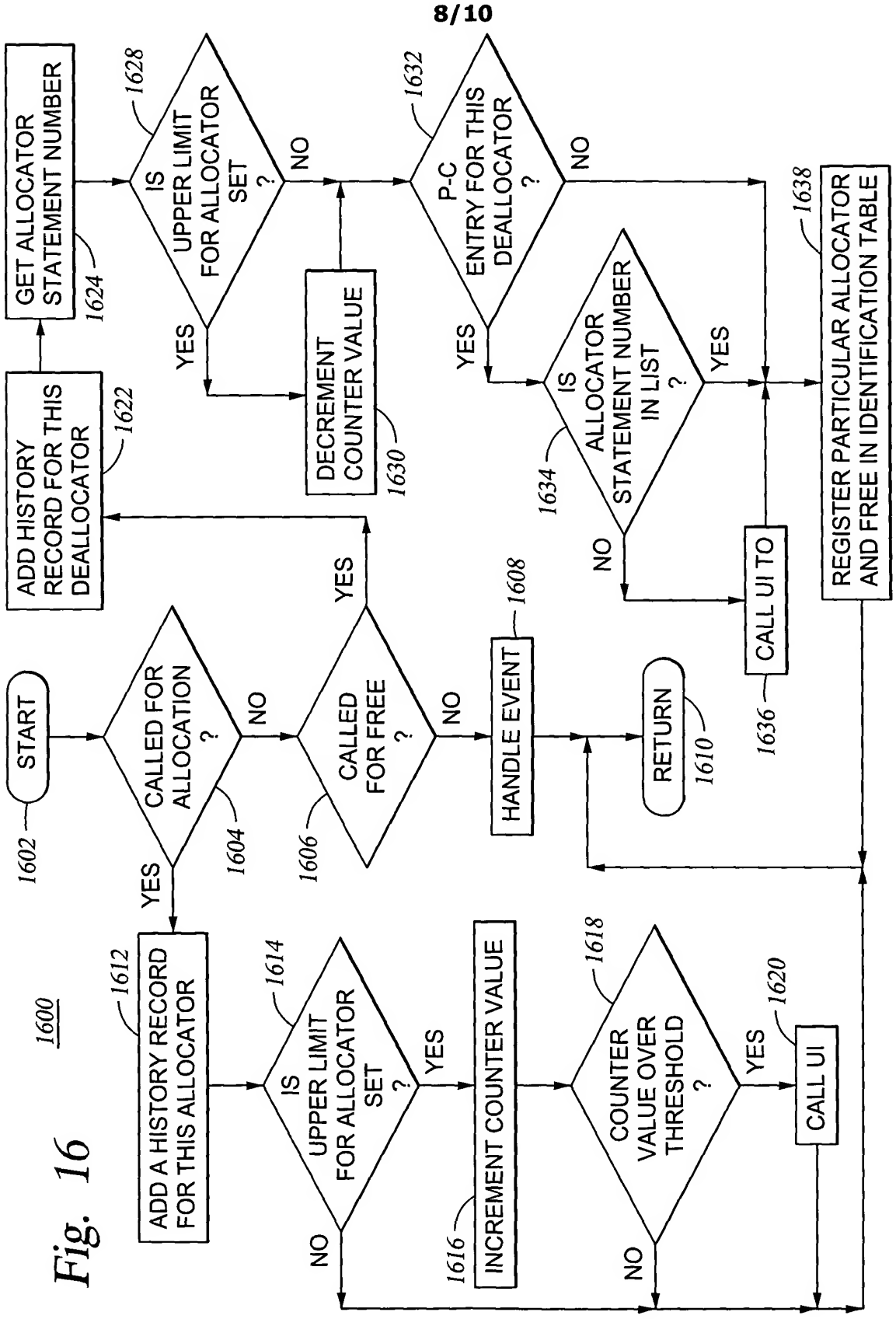
1600
START
1602
CALLED FOR ALLOCATION ?
1604
YES
NO
CALLED FOR FREE ?
1606
YES
NO
HANDLE EVENT
1608
RETURN
1610
ADD A HISTORY RECORD FOR THIS ALLOCATOR
1612
IS UPPER LIMIT FOR ALLOCATOR SET ?
1614
YES
NO
INCREMENT COUNTER VALUE
1616
COUNTER VALUE OVER THRESHOLD ?
1618
YES
NO
CALL UI
1620
ADD HISTORY RECORD FOR THIS DEALLOCATOR
1622
GET ALLOCATOR STATEMENT NUMBER
1624
IS UPPER LIMIT FOR ALLOCATOR SET ?
1628
YES
NO
DECREMENT COUNTER VALUE
1630
P-C ENTRY FOR THIS DEALLOCATOR ?
1632
YES
NO
IS ALLOCATOR STATEMENT NUMBER IN LIST ?
1634
YES
NO
CALL UI TO
1636
REGISTER PARTICULAR ALLOCATOR AND FREE IN IDENTIFICATION TABLE
1638

Fig. 16

Atty Dkt. No.: ROC920000051.D1
U.S. Serial No.: UNKNOWN
Filed: HEREWITH
Applicant: International Business Machines
Inventor: Bates et al.
Express Mail No.: EV331235171US

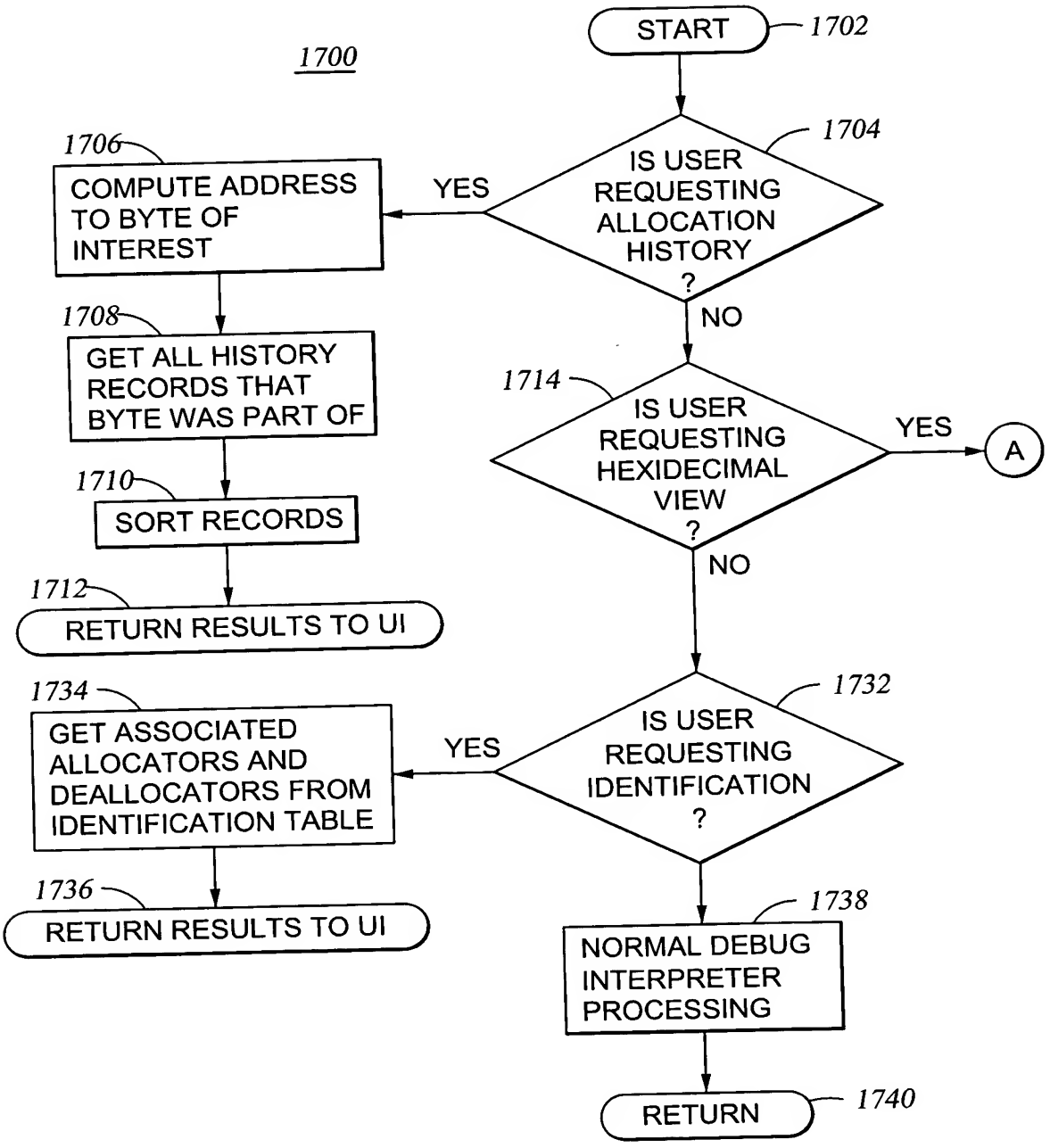

Fig. 17A

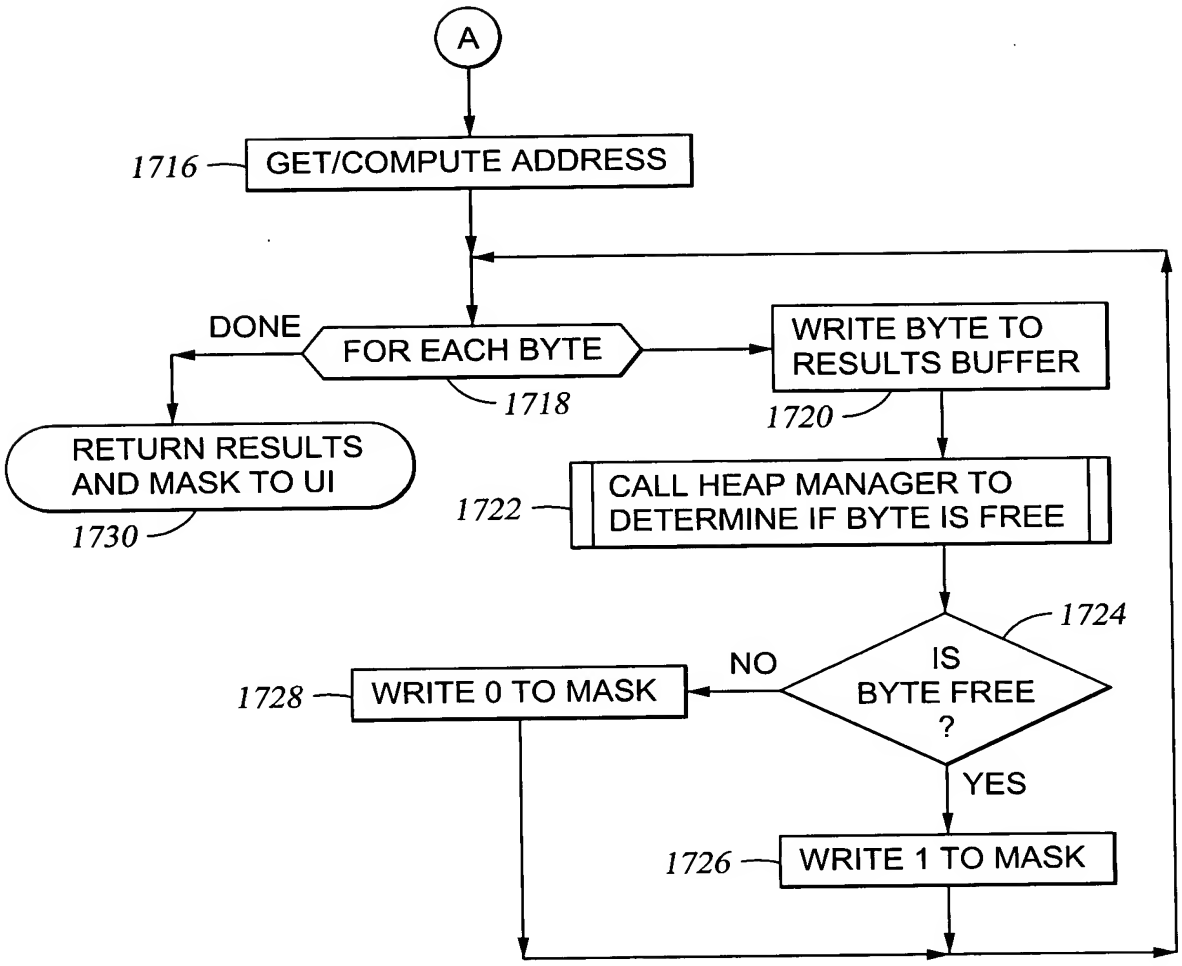
A
1716
GET/COMPUTE ADDRESS
DONE
FOR EACH BYTE
1718
WRITE BYTE TO RESULTS BUFFER
1720
RETURN RESULTS AND MASK TO UI
1730
1722
CALL HEAP MANAGER TO DETERMINE IF BYTE IS FREE
1724
IS BYTE FREE ?
NO
1728
WRITE 0 TO MASK
YES
1726
WRITE 1 TO MASK

Fig. 17B